Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 9 (2012) 1733 – 1742

Procedia
Computer Science

International Conference on Computational Science, ICCS 2012

# An Innovative Teaching Strategy to Understand High-Performance Systems through Performance Evaluation

Gonzalo Zarza[a,*], Diego Lugones[a], Daniel Franco[a], Emilio Luque[a]

[a] *Computer Architecture and Operating Systems Department, Universitat Autònoma de Barcelona, Spain*

**Abstract**

Nowadays, the study of high-performance computing (HPC) is one of the essential aspects of postgraduate programmes in Computational Science. However, university education in HPC often suffers from a significant gap between theoretical concepts and the practical experience of students. To face this challenge, we have implemented an innovative teaching strategy to provide students appropriate resources to ease the assimilation of theoretical concepts, while improving their practical experience through the use of teaching tools and resources specifically designed to promote active learning. We have used the proposed strategy to organize the module of Parallel Computers and Architectures of the Master's in High-Performance Computing, at the Universitat Autònoma de Barcelona, obtaining very promising results. In particular, we have observed improvements of both the academic marks of students and the perception about their own expertise and skills in HPC, regarding the previous teaching approach.

*Keywords:* University Education, High-Performance Computing, Simulation Management, OPNET Modeler.

## 1. Introduction

Over the last few years, high-performance computing (HPC) has turned into an important tool for modern societies, becoming the engine of an increasing number of applications and services. Along these years, the use of powerful computers has become widespread throughout many engineering disciplines. As a result, the study of parallel computer architectures is now one of the essential aspects of the academic formation of students in Computational Science, particularly in postgraduate programmes.

The Computer Architecture and Operating Systems department of the Universitat Autònoma de Barcelona offers a leading master's degree in High-Performance Computing [1], fully adapted to the European standardized university degree system [2]. Unfortunately, the education in HPC at the academic level is not exempt from challenges. In particular, postgraduate HPC courses often present significant gaps between theoretical concepts and practical experience [3], [4]. This, together with the obstacles derived from teaching people from different academic backgrounds, and the need of adapting the course content and curriculum structure to the multi-disciplinary approach of the master's studies, poses a much more complex educational challenge.

*Corresponding author. Address: CAOS Department, Campus UAB, Edifici Q, 08193 Bellaterra (Barcelona), Spain. Tel: (+34) 93 581 3533.
*Email addresses:* `gonzalo.zarza@uab.es` (Gonzalo Zarza), `diego.lugones@uab.es` (Diego Lugones), `daniel.franco@uab.es` (Daniel Franco), `emilio.luque@uab.es` (Emilio Luque)


doi:10.1016/j.procs.2012.04.191

To address these challenges, we have outlined and implemented an innovative teaching strategy to improve the learning process of students in the module of Parallel Computers and Architectures of the Master's in HPC. Innovation is given by the design and implementation of the teaching strategy around three pillars: i) comprehensive simulation models of HPC systems, to ease the assimilation of theoretical concepts; ii) a simulation management tool that allows students to configure and use several HPC systems, to improve the practical experience and the teaching of factual knowledge; and iii) a teaching method specifically designed to allow students to exploit theoretical concepts, simulation models and tools introduced along the module. This strategy is based on the know-how obtained in the master's degree over the last five years, enhanced with the research experience in HPC and parallel simulation techniques of the teaching staff.

We have observed two important aspects in the assessment our proposals. On the one hand, the improvement in the academic marks of students when applying the strategy we have outlined, as compared with the former teaching approach. On the other hand, the perception of the students about their expertise and skills at the beginning and end of the module. To this end, we have surveyed students on their knowledge in the first and the last lecture of the module in order to study differences in their own perception.

This paper is organized as follows. First, the overview of the Master's in High-Performance Computing and the module of Parallel Computers and Architectures is presented in Section 2. The proposed teaching strategy is described in detail in Section 3. Experiences and observations of the assessment are introduced in Section 4. Finally, some conclusions are drawn in Section 5.

## 2. Overview of the Master's Degree

The Master's in High-Performance Computing [1] is an Official European Master's degree of the Computer Architecture and Operating Systems department (CAOS), taught at the School of Engineering of the Universitat Autònoma de Barcelona (UAB). The master's includes scientific and technological knowledge of the members of the CAOS department, and it is oriented towards applied research and ICT companies in the information engineering and advanced computing domains.

The purpose of the master's degree is to provide appropriate scientific, technological and socio-economic training enabling students to acquire advanced skills and to conduct research tasks in computer architecture, high-performance parallel computers, artificial intelligence, software engineering, system planning and other similar fields. These studies aim at training scientists and professionals with a strong technological basis pointing to leading-edge fields such as distributed and high-performance computing. The master's also provides the theoretical and practical aspects, as well as the skills required to work in a scientific and professional level. Upon completion, students will have the scientific and methodological bases needed to develop their research skills fully in these areas. The most relevant specific and general skills are summarized in Table 1.

The modules of the master's degree are organized in a one-year programme of 60 European Credit Transfer System (ECTS) credits [5], according to the so-called Bologna process and the European Higher Education Area (EHEA) [2]. The programme consist of: i) three compulsory modules; ii) an optional module; iii) the master's research project. Each module worth 10 ECTS credits and the research project the remaining 30 ECTS credits. The complete structure of the master's programme and other additional information is available on the website of the CAOS department [1].

Table 1: Specific and general skills of the Master's in HPC.

| **1. Specific skills** |
|---|
| 1.1. Analyze and evaluate parallel computer architectures and the software developed for these systems. |
| 1.2. Investigate innovative solutions to common problems of distributed systems. |
| 1.3. Understand the last generation of network protocols, components, and structure of interconnection networks. |
| 1.4. Design solutions to complex engineering problems. |
| **2. General skills** |
| 2.1. Analyze, synthesize, organize and plan an investigation and/or research project or application. |
| 2.2. Work in multidisciplinary teams. |
| 2.3. Communicate in a clear and effective form, adapt to change, learn and take responsibility when necessary. |
| 2.4. Write and present scientific papers or technical rapports to companies or institutions. |

### 2.1. Module of Parallel Computers and Architectures

This module aims at providing students the scientific and methodological basis to: i) analyze the structure and operation of different computer systems; ii) acquire criteria for the selection of indicators and metrics to evaluate performance in HPC systems; iii) evaluate the performance of computer systems, identify the causes of possible problems, and propose strategies to improve performance.

The content of the module has been structured in eight basic blocks, as detailed in Table 2. The first block includes the description of the prevailing parallel architectures, types of parallelism, and the classification and benefits of parallel computers. Next, the second block introduces interconnection networks, notions about topology, routing and network simulations tools. Metrics and tools for performance evaluation, including benchmarks, are explained in the third block. Then, the principles of computer architecture design, and parallelism inside computing nodes are explained in blocks 4 and 5, respectively. After this, an overview of Input/Output systems is given in block 6. Topics on availability and fault tolerance are covered in block 7. At the end of the module, block 8 presents an integrated view of parallel computers. The specific skills of the module include three main topics: i) knowledge; ii) expertize; iii) attitude. The most relevant skills are summarized in Table 3.

Table 2: Structure and contents of the module of Parallel Computers and Architectures.

| **1. Parallel Computers** | **2. Interconnection Networks** | **3. Performance Evaluation** |
|---|---|---|
| - Parallel Computing Systems | - Configuration and topology | - Indexes/Benchmarks |
| - Shared Memory and Message Passing | - Routing algorithms | - Specific simulation tools |
| **4. Computer Architecture Design** | **5. Compute Node Parallelism** | **6. Input/Output** |
| - Architecture notions | - Multi-threading | - Common problems |
| - Design principles of current processors | - Multi- and many-core architectures | - Interconnection and management |
| **7. Availability and Fault Tolerance** | **8. Computing Systems** | |
| - Metrics for fault tolerance | - Memory organization and components | |
| - Reliability, prevention and redundancy | - Internal interconnects | |

Table 3: Specific skills of the module of Parallel Computers and Architectures.

| **1. Knowledge** |
|---|
| 1.1. Analyze and evaluate parallel architectures, distributed computers, and the advanced software developed for them. |
| 1.2. Analyze, design, evaluate, select and configure hardware platforms for the development and execution of applications. |
| 1.3. Use appropriate tools to analyze the performance of advanced parallel computer systems. |
| 1.4. Investigate in the field of information engineering and high-performance computing. |
| **2. Expertize** |
| 3.1. Specify, design and verify high-performance computers. |
| 3.2. Use architectural analysis techniques. |
| 3.3. Propose innovative solutions for high-performance systems. |
| 3.4. Demonstrate responsibility to manage information and knowledge, and to work in multidisciplinary projects. |
| **3. Attitude** |
| 3.1. Quality commitment, and ethical and regulatory sense. |
| 3.2. Planning and analysis capacity. |
| 3.3. Critical self criterion on assessing architectural and technological platforms. |

## 3. Teaching Strategy for High-Performance Computing

We have designed and implemented an innovative teaching strategy to reduce the gap between theoretical and practical experiences in the module of Parallel Computers and Architectures. Our proposal is based on a simple idea: provide students appropriate resources to ease the assimilation of theoretical concepts and improve the practical experience through the use of educational tools and resources designed to promote active learning, enabling students to develop the specific and general skills of the module. The objective of this strategy is to allow students to understand both the architecture and the functionality of parallel and distributed computing systems. To this end, we evaluate the performance of these systems through simulation models and benchmarking of real executions. From the analysis of these performance evaluations, students are able to understand and draw conclusions about the architecture and behavior of the systems under study. Having defined the strategy, the challenge is given by the design of adequate teaching resources and materials to reduce the separation between theory and practice.

We have organized the module of Parallel Computers and Architectures around three main pillars: i) comprehensive simulation models of parallel and distributed HPC systems; ii) a powerful simulation management tool to launch thousands of parametric simulations; iii) teaching methods specifically designed for the module. First, we have implemented a *set of simulation models* to improve the assimilation of theoretical concepts with the simulation software OPNET Modeler [6], providing students a complete overview of parallel computers and high-speed interconnection networks. Then, we have developed a *simulation management tool* [7], [8] to configure thousands of parametric simulations and launch them remotely on a variety of parallel and distributed HPC systems. Finally, we have designed the necessary *teaching method* to encourage students to acquire the advanced skills needed for conducting research tasks in computer architecture, high-performance parallel computers, and other similar fields introduced in Tables 1 and 3.

Consequently, teaching has been organized around two labs that incorporate the three pillars of the proposed strategy. In the first lab, students use the simulation models and the management tool to launch large simulations sets on a single computing system. These sets cover a wide range of parallel computer architectures that represent many different configurations of HPC systems. In the second lab, students simulate fewer systems on a wider range of real HPC systems with the simulation management tool. In the latter lab, the configured simulation scenarios play the role of benchmarking programs of real systems. An overview of the proposed strategy is shown in Fig. 1. We have designed the first lab to get across the many concepts while not necessarily having access to all the HPC systems, taking advantage of simulations. This allows students to evaluate even those systems that are not physically available and study parameters that otherwise cannot be measured. By contrast, the second lab has been designed to give students the factual knowledge needed to obtain reliable and realistic measures on real hardware and HPC systems.

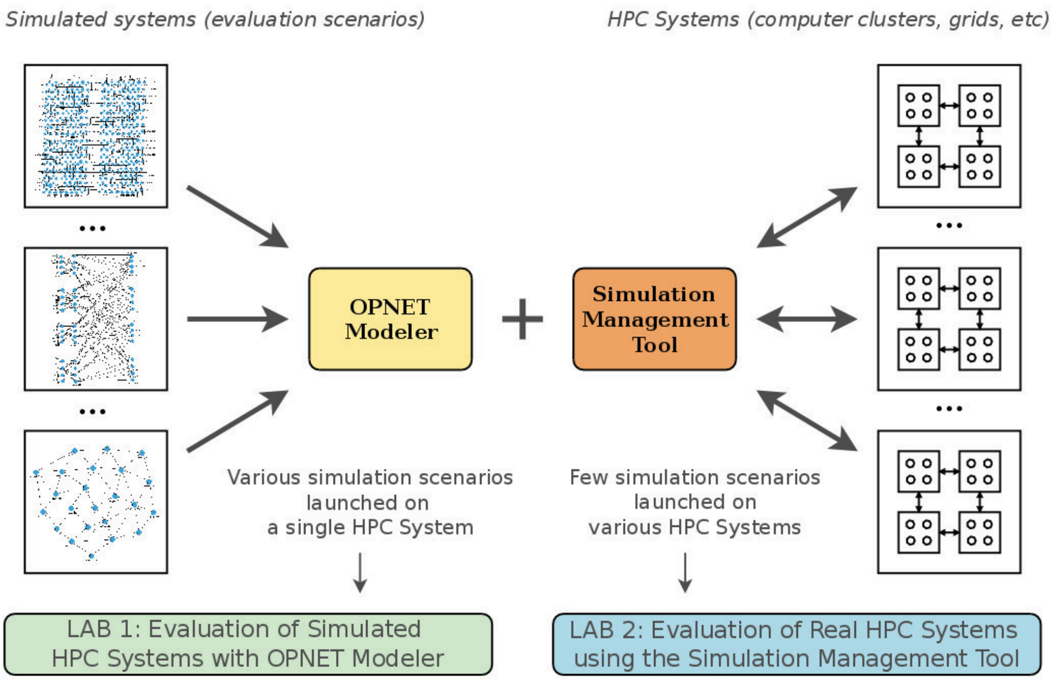
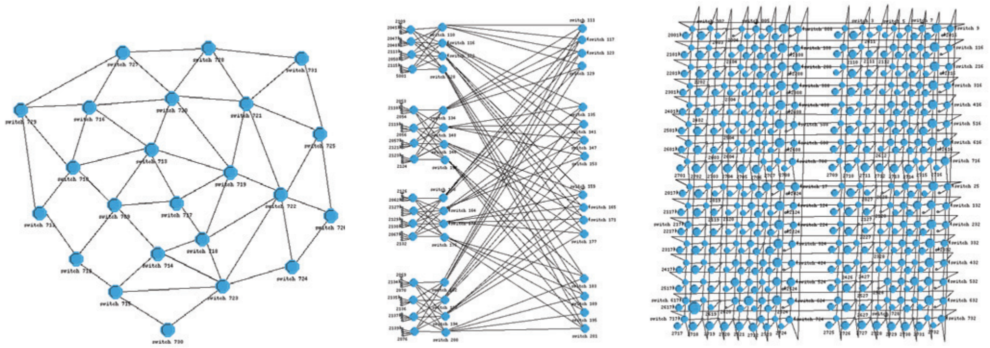



Figure 1: Overview of the teaching strategy for high-performance computing.

The following subsections explain in detail the three pillars of the proposed teaching strategy; the simulation models in Subsection 3.1; the simulation management tool in Subsection 3.2; and teaching resources in Subsection 3.3. Next, some remarks and complementary information are presented in Subsection 3.4.

### 3.1. Simulation Models for OPNET Modeler

We have implemented a complete set of models to simulate parallel computers and high-speed interconnection networks, using the standard simulation and modeling tool OPNET Modeler [6]. This model has been provided with a wide range of scenarios, designed to cover most of the contents introduced in the teaching blocks of the module (see Table 2). Some example scenarios are shown in Fig. 2. In addition, we have equipped our models with a set of configuration parameters that allow users to modify the behavior and configuration of the simulated system. Some of these parameters enable the simulation of real-based scenarios from the execution traces of real programs [9]; as well as the inclusion of failure traces of real HPC systems [10]. The most relevant parameters of the simulation models, summarized in Table 4, are: network topology; routing algorithm; traffic pattern; real-program execution traces; real-system failure traces; packet size; link frequency/speed; and router buffer size. It is clear that the combination of these configuration parameters leads to a number of simulation scenarios hardly manageable, particularly when considering different simulation seeds. For this reason, we have implemented the simulation management tool described in Subsection 3.2 that enables students to evaluate a large set of HPC systems, efficiently and orderly.



(a) Asymmetric mesh (b) Hierarchical network (c) Symmetric mesh

Figure 2: Examples of simulation scenarios based on different network topologies.

We have chosen OPNET Modeler because it provides support to model communication networks and distributed systems [11]. This tool is endowed with a three level hierarchy for modeling purposes: *network*, *node*, and *process* levels. Network level includes nodes, links, and sub-nets interconnected between them, and composing topologies. At this level, models attributes are set and parametric simulations are configured. At node level, network components are represented by using modules that typically represent applications, protocol layers, and physical resources, such as buffers, ports, and buses. Finally, the behavior of modules is programmable via their process models. They consist of finite state machines (FSM) containing blocks of C/C++ user code and kernel procedures (KP). OPNET Modeler provides a Discrete Event Simulator (DES) engine. The simulation kernel handles a single global event list and a shared simulation time clock. Events are attended from the list in the appropriate time order. Features mentioned above make OPNET a very profitable tool for modeling high-speed networks for parallel computers because it matches the tradeoff between abstraction level and simulation speed, it provides an adequate representation of real hardware, it allows the simulation of large networks, and empower the injection of real parallel application traces.

### 3.2. Simulation Management Tool

In order to enable the utilization of a wide range of real high-performance and parallel computers to run OPNET Modeler simulations, we have designed and implemented a fully functional simulation management tool [8]. This tool has been evaluated and included as a technical contribution in the industry conference *OPNETWORK 2011* [7].

We have implemented this tool to enable simulation-based mass evaluation in HPC systems that otherwise cannot be used efficiently[1], including computer clusters at the CAOS department [12], and the UAB's grid project [13]. This

[1]OPNET Modeler allows the execution of multiple discrete event simulations. However, this is only possible when using dedicated raw computing resources and most HPC installations rely on DRMs that prevent the simulation engine from directly accessing computing resources.

Table 4: Parameters of the simulation models for OPNET Modeler.

| **1. Network topologies** | **2. Routing Algorithms** | **3. Traffic Patterns** |
|---|---|---|
| - Direct | - Static | - Ad-hoc |
| - Indirect | - Dynamic | - Synthetic |
| - Hierarchical | - Application-aware | - Application-based |
| **4. Real-Program Execution Traces** | **5. Real-System Failure Traces** | **6. Additional parameters** |
| - Parallel Benchmarks | - LANL's Computing Systems | Deadlock Avoidance, Flow-Control, |
| - Scientific Applications | - PNNL's Computing Systems | Packet Size, Link Frequency, Buffer Size. |

tool enables students to launch thousands of parametric simulations to study and evaluate the complete set of systems and features provided by the simulation models introduced in Subsection 3.1. At this moment, the management tool is part of the simulation environment of different computer clusters and stand-alone machines in the CAOS department.

The main goal of the simulation management tool is to launch parametric simulations on different execution environments, such as: i) stand-alone machines; ii) non-managed computer clusters; iii) clusters relying on Distributed Resources Management (DRM) systems; iv) grid and cloud environments; and v) modern multi- and many-core architectures. To this end, we have developed an independent tool intended to act as an interface between OPNET Modeler and different execution environments. With this tool, any number of simulations can be safely sent to the above mentioned execution environments. The tool is based on a modular design that eases enhancements and layers features into two main modules: application kernel; and user interface. Both modules have been written in Java, in order to guarantee portability between different development and execution environments. Since the proposed tool is intended to interact with different systems, we have decided to use the Distributed Resources Management Application API (DRMAA) [14] for the submission and control of simulation jobs to one or more DRMs. The resulting design and the hierarchy of components and libraries could be seen in Fig. 3. The role of the management tool as the interface between OPNET Modeler and execution environments generates a number of system interactions and data-flows (shown in Fig. 3). These data-flows are:

- A first unidirectional flow between OPNET Modeler and the management tool that corresponds to the output of the simulator that constitutes the input of the management tool. The simulation models;
- A bidirectional flow between the management tool and the computing system through the DRM. On the one hand, the management tool sends one or more parametric simulations to the corresponding computing system. On the other hand, the DRM sends back status information about queue jobs and the results of correctly executed simulations (to the management tool).

The user interface of the management tool is composed by a number of tabs specifically distributed to show related information in an ordered manner, as shown in the snapshot of Fig. 4(a). The tool has been successfully tested on two executions environments, GNU/Linux and Microsoft Windows (see Fig. 4(b)). At this moment, the graphical interface of the visualization module relies on Java Swing components but the modular design of the tool allows easy transitions to web-based user interfaces.

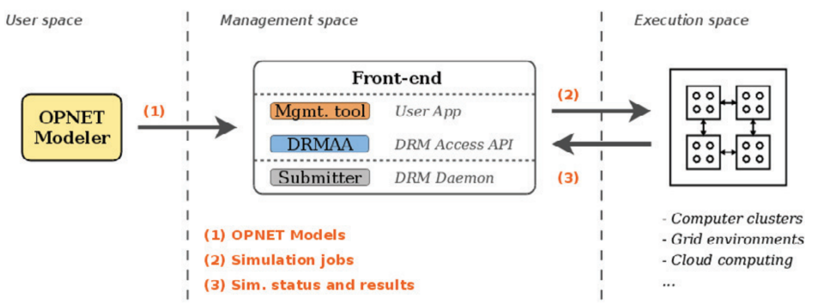
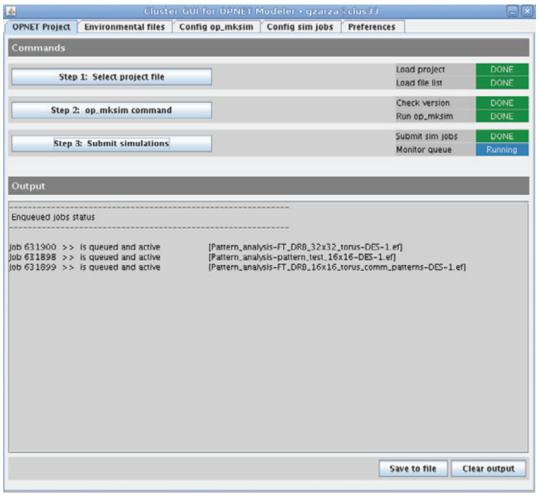
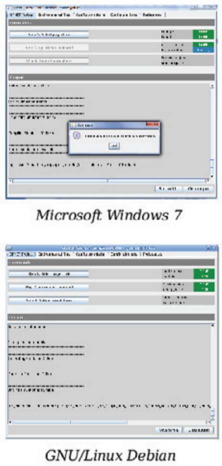



Figure 3: Overview of the simulation management tool.

The management tool has been designed to very simple and usable. This has been achieved by reducing the critical operations to three simple and incremental steps: i) the selection of the simulation model; ii) the generation of one or more parametric simulation job(s), as appropriate; and iii) the submission of simulations to the execution space/environment. These steps are required for safely compiling and sending parametric simulations to different execution environments. Anytime a simulation job ends, the management tool writes two output files in the user's home directory. One file contains the errors reported by the DRM during the execution of the simulation (if any); while the other keeps a record of the simulation output. If no errors occur, OPNET vector files are properly generated. These output files contain simulation results and can be visualized with the standard tools of OPNET Modeler, like the files generated from the simulation engine itself.




(a) Main view of the tool.




(b) Execution examples.

Figure 4: Implementation of the simulation management tool.

### 3.3. Teaching Resources

Educational resources have been designed to provide appropriate scientific and technological training to students, enabling them to acquire advanced skills to conduct research tasks in the fields of computer architecture and high-performance parallel computers. To this end, teaching has been organized around two labs that incorporate both the general skills of Master's in HPC and the specific skills of the module of Parallel Computers and Architectures.

The first lab has been designed to enable students to analyze and evaluate parallel architectures, and distributed computers (skill 1.1 in Table 3). This lab consists on the study and utilization of the simulation models presented in Subsection 3.1 and the management tool introduced in Subsection 3.2 to assimilate the theoretical concepts of the module. First, students are required to study and configure various simulation scenarios, modifying some of the parameters listed in Table 4. Once the simulation scenarios have been correctly configured and presented to the teaching staff, students must set-up the simulation management tool to launch parametric simulations on a specific HPC system. In this first stage of the learning process, students are allowed to use a computer cluster dedicated exclusively to teaching purposes, the Cluster A (detailed in Table 5).

In addition, we have designed a second lab to allow students to: evaluate and configure hardware platforms; and use appropriate tools to analyze the performance of advanced HPC systems (skills 1.2 and 1.3 in Table 3). This lab is intended to strengthen theory through practice by allowing students to configure and use a wider variety of real HPC systems. To this end, in this second lab students are requested to use the simulation management tool to launch simulations in all four computing systems presented in Table 5: the Cluster A; two computer clusters commonly used for research purposes at the CAOS department, Clusters B and C; and some distributed computing resources belonging to the School of Engineering [13], accessed by means of a workload management system (Condor).

Finally, to develop the general skills of the master's programme, students are required to successfully accomplish three milestones: i) work in groups of two or three people to prepare the labs, depending on the number of students enrolled; ii) write a short paper explaining the results obtained at the end of each lab; and iii) prepare a 15-minute lecture to expose the results of both labs at the end of the module.

Table 5: Computing systems used in Labs.

| **1. Cluster A (Teaching)** | **2. Òliba Computers (Grid Project UAB)** |
|---|---|
| Quad-Core Intel(R) i7 CPU 950, 3.06 GHz, 256 KB L2, 8 MB L3, 6 GB RAM Fully Buffered DIMMS 667 MHz, Network Gigabit Ethernet, 40 cores (10x4). | Intel(R) Pentium IV, Northwood CPU, 2.8 GHz, 256 KB L2, 512 MB RAM, Network Gigabit Ethernet, 20 cores (20x1) at the School of Engineering |
| **3. Cluster B (Research)** | **4. Cluster C (Research)** |
| Dual-Core Intel(R) Xeon(R) CPU 5150, 2.66 GHz, 4 MB L2, 8 GB RAM Fully Buffered DIMM 667 MHz, Network Gigabit Ethernet, 128 cores (32x4). | 2 x Quad-Core Intel(R) Xeon(R) CPU E5430, 2.66 GHz, 2x6 MB L2, 16 GB RAM Fully Buffered DIMMs 667 MHz, Network Gigabit Ethernet, 64 cores (8x8). |

### *3.4. Final Remarks*

The biggest strength of the proposed teaching strategy is the simplicity of the *three-pillar* approach which enables the adoption of the strategy in other programmes, with little effort and no additional monetary cost. This, together with the soundness of the innovative tools and teaching methods we have presented, constitute the novelty and the main contribution of this work[2]. It is worth noting that at least one student per year has used the proposed simulation models and the management tool in the Master's Thesis [15], [16], [17] since the strategy has been applied to the module of Parallel Computers and Architectures. These are encouraging results.

## 4. Experiences and Observations

A careful and detailed analysis and review of short papers and lectures presented by students at the end of the module illuminated some key insights as to how students have been able to improve their analysis and research skills. For instance, we could infer that most students are able to design and conduct experiments using real systems to simulate real-based scenarios. In addition, some students have also showed interest on conducting complex and elaborated performance evaluations. This initiative of students indicates the existence of real benefits arising from the application of our teaching strategy.

At the beginning and end of the module of Parallel Computers and Architectures, students are surveyed on their experience with different aspects of the module and on the labs in particular[2]. Over the last six academic years, we have surveyed all students of the module (68 people). In the last three courses, we have observed an improvement of up to 29% in the average academic mark of students in the module, as could be seen in Fig. 5. These improvements coincide with the implementation of our proposal in the academic year 2009/2010. It is worth noting that academic marks were quite good for a master's degree before the implementation of our proposal, therefore, the improvement achieved represents a major enhancement. The satisfaction level of students regarding the balance between theory and practice, and the applicability of concepts have increased a 36% (Fig. 6(a)) and 37% (Fig. 6(b)), respectively.

To explore the students' own perception about their expertise and skills, we have compared their responses on the surveys presented at the beginning and end of the module. Due to space constraints, we only present results for the academic year 2011/2012 in this paper. The most important outcomes could be seen in Figs. 7(a) and 7(b). Overall, when applying our strategy over than 66% of students felt that the explanation of the simulation process was understandable. This fits with our goal of enabling students to acquire advanced analysis skills. It is also clear from results of Fig. 7(a) that most students had little or no knowledge about performance evaluation techniques at the beginning of the module, which gives even more value to the above findings. At the end of the module (Fig. 7(b)), around 99% of students still consider the possibility of launching hundreds of parametric simulations *a must* when evaluating HPC systems. Over 50% of the students felt like they learned from labs using both OPNET models and the simulation management tool, and consider themselves capable of applying these resources in the Master's Thesis.

[2] The set of simulation models, management tool, lab assignments, and surveys can be accessed at `http://caos.uab.es/~gzarza/`.

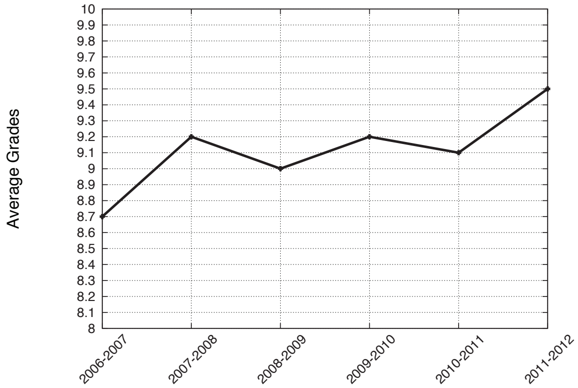
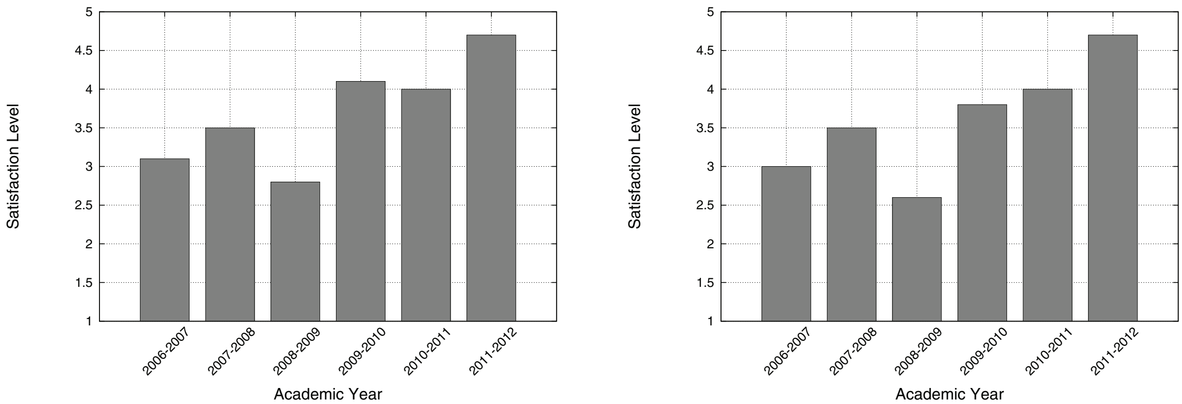
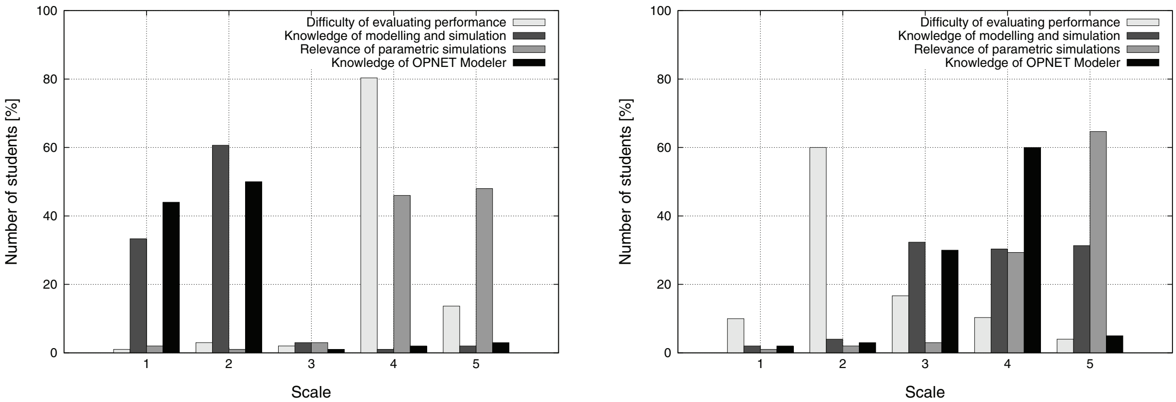



Figure 5: Average grades of students per year. Master's in HPC, 2006/2007 to 2011/2012.




(a) Theory vs. Practice (1: poor; . . . ; 5: excellent) (b) Applicability of contents (1: poor; . . . ; 5: excellent)

Figure 6: Satisfaction level of students. Master's in HPC, 2006/2007 to 2011/2012.




(a) At the beginning of the module (1: low; . . . ; 5: high) (b) At the end of the module (1: low; . . . ; 5: high)

Figure 7: Student surveys results. Master's in HPC, 2011/2012.

## 5. Conclusions

In this paper, we have presented an innovative strategy for teaching High-Performance Computing in postgraduate programmes, designed and implemented to improve learning in the Master's in High-Performance Computing of the Universitat Autònoma de Barcelona. Innovation is given by the application of a simple but effective organization the module of Parallel Computers and Architectures around three main pillars:

- *A comprehensive set of simulation models,* to provide students the overview of parallel computers and high-speed interconnection networks, covering most of the theoretical concepts introduced in the module.
- *A powerful simulation management tool,* to strengthen theory through practice by allowing students to configure and use a variety of real high-performance computing systems to launch parametric simulations.
- *Appropriate teaching resources,* specifically designed to encourage students to acquire advanced skills to conduct research tasks in computer architecture, high performance parallel computers, and other similar fields.

The implementation of the proposed strategy in the master's degree has yielded encouraging results. Particularly, since we began to implement the proposed strategy in the academic year 2009/2010, we have observed a steady improvement in two aspects closely related to the quality of teaching: i) academic marks of students; and ii) the perception of students about the expertise and skills they have acquired along the module. These positive results demonstrate the benefits and effectiveness of the proposed teaching strategy.

We are currently working on the improvement of the simulations models to enable accurate evaluation of novel processors, including multi- and many-core architectures. We expect to include these enhancement in the next course (academic year 2012/2013).

## Acknowledgments

This research has been supported by the MICINN-Spain under contract TIN2007-64974. We thank OPNET Technologies for the licenses used for academic research and teaching in the Master's in High-Performance Computing.

## References

[1] CAOS Department, Official Master's Degree High Performance Computing (2011).
URL `http://www.caos.uab.es/mscphd.php?id=1`

[2] B. Wachter, The Bologna Process: developments and prospects, European Journal of Education 39 (3) (2004) 265–273.

[3] M. Bernreuther, M. Brenk, H.-J. Bungartz, R.-P. Mundani, I. Muntean, Teaching high-performance computing on a high-performance cluster, in: Computational Science ICCS 2005, Vol. 3515 of LNCS, Springer Berlin / Heidelberg, 2005, pp. 251–256.

[4] A. Georgi, S. Hohlig, R. Geyer, W. E. Nagel, Linux Cluster in Theory and Practice: A Novel Approach in Teaching Cluster Computing Based on the Intel Atom Platform, Procedia CS 4 (2011) 1917–1926.

[5] E. Communities, ECTS Users' Guide, European Commission Publications, 2009. `doi:10.2766/88064`.

[6] OPNET Technologies, OPNET Modeler Accelerating Network R&D (2011).
URL `http://www.opnet.com/`

[7] C. Nuñez, G. Zarza, D. Lugones, J. Navarro, D. Franco, E. Luque, ClusterGUI, an Application to Launch OPNET Simulations within Resource Managed Environments, in: OPNETWORK Conference, 2011, pp. 1–7.

[8] G. Zarza, D. Lugones, D. Franco, E. Luque, A Tool for Enabling Simulation-Based Mass Evaluation of the Internet of Things in High-Performance Computing Systems, Submitted to the journal Simulation Modelling Practice and Theory (January 2012).

[9] A. Wong, D. Rexachs, E. Luque, Extraction of parallel application signatures for performance prediction, in: HPCC, 2010, pp. 223–230.

[10] G. Zarza, D. Lugones, D. Franco, E. Luque, Fault-tolerant Routing for Multiple Permanent and Non-permanent Faults in HPC Systems, in: Proceedings of the Intl. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA), 2010, pp. 144–150.

[11] D. Lugones, D. Franco, E. Luque, Modeling Adaptive Routing Protocols In High Speed Interconnection Networks, in: OPNETWORK Conference, 2008, pp. 1–7.

[12] CAOS Department, Research Clusters (2011).
URL `http://caos.uab.es/resources.php?lang=en`

[13] Universitat Autònoma de Barcelona, Projecte Òliba (2011).
URL `http://www.oliba.uab.es/`

[14] ORG DRMAA Working Group, DRMAA: Distributed Resources Management Application API (2011).
URL `http://www.drmaa.org/`

[15] C. Nuñez, Predictive and distributed routing balancing (PR-DRB): high speed interconnection networks, Master's thesis, Universitat Autònoma de Barcelona (2010).

[16] A. Gómez, Model of Interconnection Networks based on Links, Master's thesis, Universitat Autònoma de Barcelona (2010).

[17] G. Chaín, Analytical Model for Estimating the Latency in High-Speed Networks, Master's thesis, Universitat Autònoma de Barcelona (2011).